DOCUMENT RESUME

ED 331 465 IR 013 791

AUTHOR Collins, Allan
TITLE Cognitive Apprenticeship and Instructional Technology. Technical Report.
INSTITUTION BBN Labs, Inc., Cambridge, MA.
SPONS AGENCY Office of Naval Research, Arlington, Va. Personnel and Training Research Programs Office.
REPORT NO BBN-R-6899
PUB DATE Jan 88
CONTRACT N00014-85-C-0026
NOTE 34p.; For a related report, see IR 013 792.
PUB TYPE Viewpoints (Opinion/Position Papers, Essays, etc.) (120) -- Reports - Research/Technical (143)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Cognitive Processes; *Computer Assisted Instruction; *Educational Environment; *Educational Technology; Models; Problem Solving; *Teaching Methods; Tutoring
IDENTIFIERS *Cognitive Apprenticeships; *Situated Learning

ABSTRACT

In earlier times, practically everything was taught by apprenticeships. Schools are a recent invention that use many fewer teaching resources, but the computer enables us to go back to the resource-intensive mode of education, in a form called cognitive apprenticeship. This involves the use of modeling, coaching, reflecting on performance, and articulation methods of traditional apprenticeships, but with an emphasis on cognitive rather than physical skills. In the situated learning approach, knowledge and skills are taught in contexts that reflect how the knowledge will be used in real life situations. Technology thus enables us to realize apprenticeship learning environments that were either not possible or not cost effective before. (3 figures, 31 references) (EW)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

BBN Systems and Technologies Corporation

A Subsidiary of Bolt Beranek and Newman Inc.

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Report No. 6899

ED331465

Cognitive Apprenticeship and Instructional Technology

Technical Report

Allan Collins

January, 1988

Approved for publication; distribution unlimited

IR013791

BEST COPY AVAILABLE

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) BBN Report #6899	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Bolt Beranek and Newman 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Personnel and Training Research Programs Office of Naval Research (Code 1142PT)
6c. ADDRESS (City, State, and ZIP Code) 10 Moulton Street Cambridge, MA 02238	7b. ADDRESS (City, State, and ZIP Code) 800 North Quincy Street Arlington, VA 22217-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-85-C-0026
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO: 61153N; PROJECT NO: RR04206; TASK NO: RR04206-0A; WORK UNIT ACCESSION NO: NR 667-540

11. TITLE (Include Security Classification)

Cognitive Apprenticeship and Instructional Technology

12. PERSONAL AUTHOR(S)
Allan Collins

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical Report	FROM 1/1/87 TO 6/30/88	January, 1988	16

16. SUPPLEMENTARY NOTATION

To appear in B.F. Jones and L. Idol (Eds.) Dimensions of THinking and Cognitive Instruction Hillsdale, NJ:Lawrence Erlbaum Associates.

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Education, Teaching, Thinking, Computers and Education, Learning, Educational Technology

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

In earlier times, practically everything was taught by apprenticeship: growing crops, running trades, administering governments. Schools are a recent invention that use many fewer teaching resources. But the computer enables us to go back to a resource-intensive mode of education, in a form we call cognitive apprenticeship (Collins, Brown & Newman, in press). As we argue in the earler paper, cognitive apprenticeship employs the modeling, coaching, and fading paradigm of traditional apprenticeship, but with emphasis on cognitive rather than physical skills. The basic thesis in this paper is that technology enables us to realize apprenticeship learning environments that were either not possible or not cost effective before.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. Susan Chipman	22b. TELEPHONE (Include Area Code) 202-696-4318 22c. OFFICE SYMBOL ONR 1142PT

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Cognitive Apprenticeship and Instructional Technology

Allan Collins
BBN Laboratories Incorporated
Cambridge, MA 02238

This research was supported by the Personnel and Training Research Programs, Psychological Sciences Division, Office of Naval Research under Contract No. N00014-C-85-0026, Contract Authority Identification Number, NR 667-540.

To appear in B.F. Jones and L. Idol (Eds.) Dimensions of Thinking and Cognitive Instruction. Hillsdale, NJ: Lawrence Erlbaum Associates.

Cognitive Apprenticeship and Instructional Technology

Allan Collins

In earlier times, practically everything was taught by apprenticeship: growing crops, running trades, administering governments. Schools are a recent invention that use many fewer teaching resources. But the computer enables us to go back to a resource-intensive mode of education, in a form we call cognitive apprenticeship (Collins, Brown & Newman, in press). As we argue in the earlier paper, cognitive apprenticeship employs the modelling, coaching, and fading paradigm of traditional apprenticeship, but with emphasis on cognitive rather than physical skills. My basic thesis in this paper is that technology enables us to realize apprenticeship learning environments that were either not possible or not cost effective before.

This paper addresses the questions: what kind of leverage do we derive from computer technology, and what design criteria can we specify for building computational learning environments? We have developed a tentative set of characteristics (Collins, Brown & Newman, in press) we think computational learning environments should have, based on analyzing what kinds of tutoring systems we see emerging, what we have learned from studies such as Lampert (1986) Palincsar and Brown (1984), Scardamalia, Bereiter and Steinbach (1984), and Schoenfeld (1983, 1985), and what resource rich learning environments (such as tennis coaches and graduate school instruction) are like.

This paper discusses six characteristics of cognitive apprenticeship for which technology provides particular leverage. For each abstract characteristic I will address:

a) What the abstraction refers to,

b) The implications technology has for realizing the abstraction in practice,

c) Why realizing the abstraction is of benefit to students,

d) An example of a computer system that embodies the abstraction.

1 Situated Learning

Situated learning is the notion of learning knowledge and skills in contexts that reflect the way the knowledge will be useful in real life. It is the sine qua non of apprenticeship. But it should be thought of in the most general way. In the context of math skills, they might be taught in contexts ranging from running a bank, or shopping in a grocery store, to inventing new theorems or finding new proofs. That is, situated learning can incorporate situations from everyday life to the most theoretical endeavors (Schoenfeld, in press).

The computer allows us to create environments that mimic situations in the real world that we cannot otherwise realize in a classroom (or home). One approach is through microworlds, but also through computer networks, data bases, graphing packages and text editors (Collins, 1986). There are inherent dangers in microworlds, such as learning to make decisions based on a few variables rather than a rich set of variables (Dreyfus and Dreyfus, 1986), but potentially the benefits far outweigh the risks.

The benefits of situated learning include:

a) Students learn conditions for applying knowledge. By learning arithmetic, for example, in representative contexts such as grocery shopping or running a bank, the student ties the knowledge learned to specific contexts. Then when they are in novel situations, e.g. buying airline tickets or working in an accounting department, they will be able to see how the knowledge learned might apply in these new situations by analogy to the situations they learned about.

b) Situations foster invention. When students use their knowledge to deal with real problems and situations, they are forced to make inventions to apply their knowledge (Lampert, 1986). Thus they are learning how to use their knowledge flexibly to deal with novel situations.

c) Students see the implications of the knowledge. When learning is embedded in context, then its uses are more apparent to students. They can actually see how the knowledge is used in different settings, and what power it gives them to use the knowledge. It is not readily apparent to students how most of what they learn in school might be used.

d) Context structures knowledge appropriate to its uses. When students learn things for school, they often invoke suboptimal schemes for remembering the information. For example, they may infer that all arithmetic word problems with the word "left" (e.g., "Mary had 7 apples. She gave 3 to John. How many did she have left?") are subtraction problems (Schoenfeld, in press). Or they may memorize facts in a rather rote way so that they can be retrieved for a test (e.g., the capital of Oregon is Salem: with the oar in Oregon you can go sailing). When knowledge is

learned in the contexts of its uses it is more likely to be stored in a form that is usable in novel contexts.

These may be the central issues as to why transfer of knowledge is so difficult: by learning in multiple contexts you learn different ways knowledge can be used, and begin to generalize on these ways. This is opposed to the way we teach knowledge in an abstract way in schools now, which leads to strategies, such as depending on the fact that everything in a particular chapter uses a single method, or storing information just long enough to retrieve it for a test. Instead of trying to teach abstract knowledge and how to apply it in contexts (as word problems are supposed to do), we advocate teaching in multiple contexts and then trying to generalize across those contexts. In this way knowledge becomes both specific and general.

An excellent example of a computer-based situated learning environment based on a microworld is Geography Search by Tom Snyder (Snyder & Palmer, 1986) which is one of five programs in the Search Series by McGraw Hill. It teaches history, math, planning, and problem solving. In this simulated microworld groups of students sail ships from Europe to the New World about the time of Columbus, in order to look for a treasure that is distributed around North and South America. Land and other ships come into view on the screen when the ship nears them. Students have to calculate their route using sextant and compass in the way sailors did of old. They must also keep track of food and supplies, so then don't run out while they are at sea. So students are learning history and math in a context where novel problems continually arise.

Another example of situated learning is the reconvening of the Constitutional Convention among school students in the Boston area using a computer mail system (personal communication, William Fitzhugh). Different schools represent different delegations (e.g. Delaware, Virginia). The students prepare by reading about the concerns of their states in 1787. During the convention they will try to negotiate a draft constitution to correct for the difficulties that were encountered with the Articles of Confederation. A similar kind of convention could be held to cope with the modern day problems that have arisen with the Constitution (e.g., the budget process, the advent of media and its expenses, the disagreements over who controls foreign policy, the difficulties when Presidents are disabled for any reason). Government and its structures become real in dealing with these kinds of questions.

History is typical of the information schools teach in a non-situated way. Schools try to pour in a lot of facts and theories, and make no use of that knowledge other than recall. Computers give us enormous power to create situated learning environments where students are learning about reading, writing, math, science, and social studies in ways that reflect the kinds of activities they will need these for.

2 Modelling and Explaining

Modelling is showing how a process unfolds and explaining involves giving reasons why it happens that way. It is the showing and telling that is so characteristic of apprenticeship.

Two kinds of modelling are important for education:

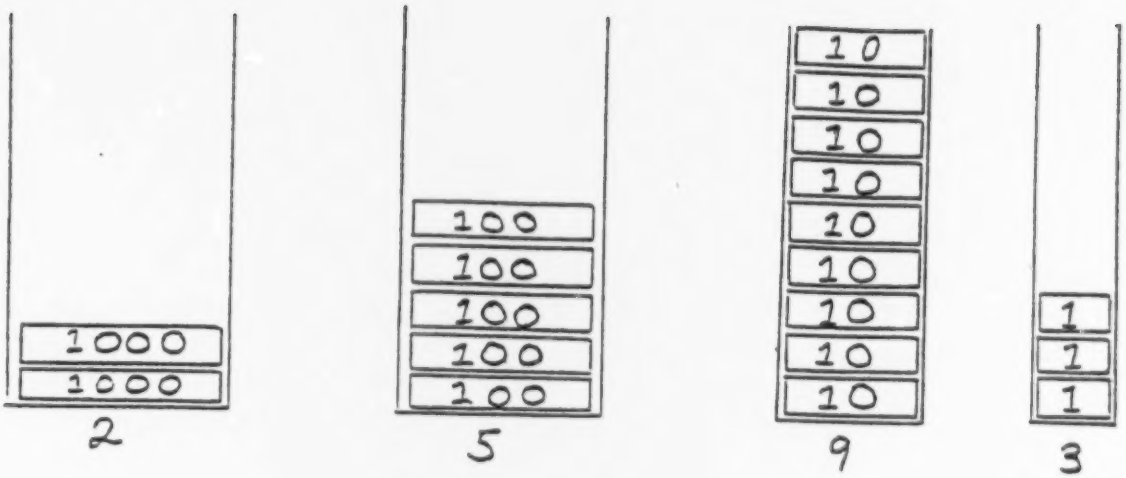
1. Modelling of processes in the world: For example, one might show how electrons move in circuits (Haertel, 1987), or how information coded in DNA is translated into protein molecules.

2. Modelling of expert performance: For example, in teaching reading, the teacher might read in one voice and verbalize her thinking in another voice, like a slow motion movie. She could verbalize what is confusing, what to do when you don't understand, any tentative hypotheses about what is meant and what will come later, any evidence as it comes in about these hypotheses, her summaries and integrations, her guesses as to the author's intentions, and evaluations of the structure and style of the writing: in short, all the thoughts of a skilled reader (Collins and Smith, 1982).

The computer makes it possible to represent processes in ways books never could, and even in ways people cannot. Computers can make the invisible visible: they let you see inside pipes or inside the body, how current changes in circuits based on electron flow, where the center of mass for a group of bodies is, how microscopic processes unfold. At the same time they can make tacit knowledge explicit, by showing the strategies experts use to solve problems that students set for them. To the degree we can develop good process models of expert performance, we can embed these in technology, where they can be observed over and over for different details.

Computers can use multimedia, i.e. animation, voice, text and graphics to characterize different aspects of processes. Ways of integrating animation and voice are just beginning to be explored, but it is clear that they have enormous potential for making things clearer. For example, it is possible to highlight each component of a system while it is talked about. One can show both what happens and what does not happen. We can render unto voice what verbal description best transmits (e.g. reasons why, abstract ideas) and render unto animation what visual description best transmits (e.g., processes and relations between components, concrete ideas). And we can achieve simultaneous presentation so that what is seen happening on the screen can be explained orally without looking elsewhere. Eventually much of the information in text books and libraries will be in this form.

The benefits of modelling include:



$$\begin{array}{r} {\scriptstyle 11} \\ 2593 \\ \underline{\quad\;\; 9} \\ 2602 \end{array}$$

Figure 1. Bin model representation of 2593 and standard representation of the addition problem 2593 + 9 in Summit (from Feurzeig & White, 1984).

a) Seeing expert solutions to problems set by the student. For the most part in school, students never see how experts solve problems--they see only worked out solutions. Worked out solutions do not show the false starts or dead ends that characterize real world problem solving. If students can pick problems that raise issues in their mind, and watch how an expert computer program attacks the problem, then they will have a genuinely new kind of learning experience.

b) Integrating what happens and why it happens. Demonstrations in class or process descriptions in text have inherent limitations. Many demonstrations occur too fast to assimilate what is happening and why it is happening. In books simultaneity can only be approximated with static diagrams and text explanation. The ability to see what is happening, in slow motion if need be, and hear at the same time a verbal explanation of why it is happening facilitates building an integrated understanding of processes.

c) Making visible parts of a process not normally seen. Because much of what we want students to learn is not ordinarily seen, students must infer from end products and a few intermediate states how processes unfold. By revealing these processes in detail, many more students will have a chance to figure out what is happening.

I can illustrate modelling with a computer system called Summit (Feurzeig & White, 1984), designed for teaching addition and subtraction. Summit combines visual animation with spoken explanation. The system had two representations for addition and subtraction: the standard algorithm and a bin-model representation, derived from Dienes (1960) blocks. In both representations students can pose problems to the system to see what happens. Figure 1 illustrates how the system represents a simple addition problem (2593+9) in the two representations.

The bin model works as follows. Suppose that the ones bin has 3 icons in it and 9 more are added to it. The model first displays all 12 icons, stacked one on top of the other, in the ones bin. Next, the computer says, "Now there are too many ones. We have to take some to the tens." An empty box then appears in between the ones and tens bins. Next 10 icons are removed from the ones bin one at a time and placed in the box. There is a counter displayed on the box. When there are 10 icons in the box, the box becomes a tens icon, which is added to the tens bin. This in turn leads to an overflow in the tens bin, and the process repeats itself.

Summit models the addition process for the standard algorithm in a similar way, saying: "Then we add the ones. 3 plus 9 is 12, but 12 won't fit. So we write the 2 under the ones column, and take the 1 over to the tens column," writing it just above the tens column. "Then we add the tens. 1 plus 9 is 10, but 10 won't fit. So we write 0 in the tens column, and take the 1 over to the hundreds column," and so on. Each of the actions appears on the screen as the voice explains it.

There are other systems such as Sophie (Brown, Burton & deKleer, 1982) and Quest (White & Frederiksen, in press) in the electricity domain that not only model how a process unfolds, but how experts troubleshoot a faulty system. In these systems students can see an expert troubleshooting whatever faults they decide to set in the system. Thus, they can pick problems that are at the edge of their own competence in order to see how to extend their troubleshooting strategies.

3 Coaching

Computer systems have the ability to patiently observe students as they try to carry out tasks, providing hints or assistance when needed. This kind of personal attention is simply not feasible in most school classrooms.

Not only is the computer patient, it can remember perfectly what the student did before. It can consider multiple hypotheses about the difficulties the student is having (Burton, 1982), and it can observe over a period of time in order to tell what problems the student is really having. Moreover, a computer coach gives students a different perspective from which to understand their own performance.

The benefits of coaching include:

a) Coaching provides help directed at real difficulties. By observing students in a problem solving situation, the coach can see what difficulties a particular student is having. Because teachers in school rarely have a chance to observe student problem solving, most of the help they give is not really directed at the problems students actually have.

b) Coaching provides help at critical times. Coaching provides help to students when they most need it: when they are struggling with a task and are most aware of the critical factors guiding their decisions. Thus they are in the best position to use any help given.

c) Coaching provides as much help as is needed to accomplish tasks. Coaching enables students to do tasks they might not otherwise be able to complete. It gives them a sense that they can really do difficult tasks. As they become more skilled, the coach's role can fade giving students more and more control over execution of the task.

d) Coaching provides new eyeglasses for the student. Coaching can help students see the process from an entirely different perspective. The coach can point out things that do not go as expected and explain why. Furthermore, a coach can introduce new terms such as "forward chaining" and "thrashing" (see below) that are critical to employing heuristic and metacognitive strategies.

The best example of a computer coach is the coach built by Burton and Brown (1982) for the computer game "How the West Was Won." Originally developed as part of the Plato system, the game, which is a variant on Chutes and Ladders, is designed to teach children basic arithmetic operations. It can be played either by one person against the computer or by two people against each other. Figure 2 shows a display of what the game board looks like to the players.

The rules of the game are as follows: When it is a person's turn, they must form an arithmetic expression from the three spinners using two different operations (e.g. 2x1+2), and after they have formed such an expression they must input the value of the expression (in this case 4). If they miscalculate, they lose their turn; if they are correct, they move forward on the board the number of spaces calculated. The object is to reach the last town (which is 70) before your opponent, but to reach the last town you must form an expression that lands you exactly on the town (i.e. you cannot overshoot it). There are some special rules that make the game more challenging: a) if you land on a town (every 10 squares) you advance to the next town, b) if you land at the beginning of one of the short cuts (i.e. 5, 25, 44) you advance to the end of the shortcut (i.e. 13, 36, 54), and c) if you land on your opponent you send her back two towns.

These special rules make it advantageous to consider many possible different moves, which would require trying out different arithmetic expressions and calculating their values. But students playing the Plato game do not consider alternative moves: they tend instead to lock onto a particular strategy, such as multiplying the largest number times the sum of the other two numbers, which gives them the largest value they can make. Thus they do not play very well or learn much arithmetic without coaching.

To remedy this situation Burton & Brown (1982) built a computer coach that observes students as they play the game and gives them hints or advice at critical moments. The computer coach rank orders every possible move a player might make given the three spinner values. The rank order is constructed with respect to how far ahead of your opponent any move leaves you. This gives the coach a way to evaluate the effectiveness of any move.

The coach observes the players moves by looking at certain "issues" e.g. whether the student knows to land on towns or on the opponent when it is effective to do so, whether the students knows to use parentheses or the minus sign or the divide sign when it is effective to do so. If the student systematically fails to make moves that require understanding of any of these issues, then the coach will notice the pattern. The coach will then intervene with a hint about a particular issue when that issue is particularly salient to making a good move; i.e. when the student's move is much inferior to the best move possible taking into account that issue. The intervention by the tutor occurs just after the student's move: the system points out how using

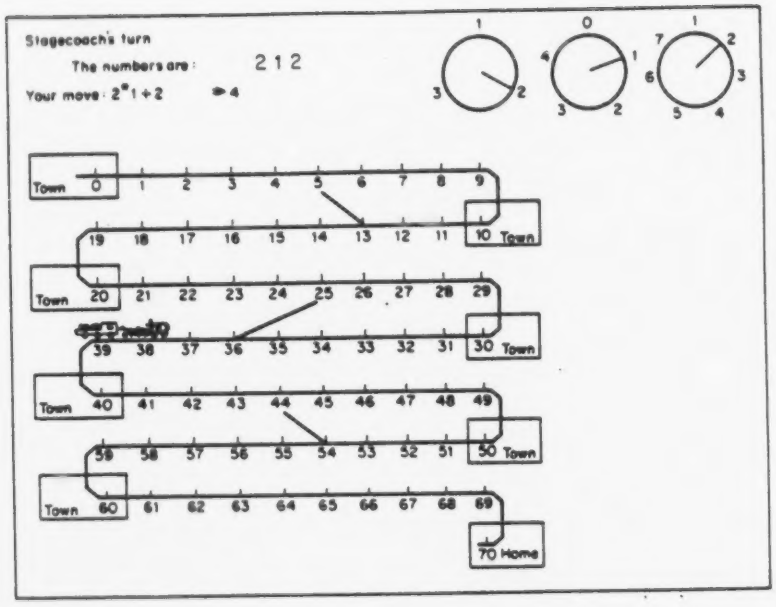


Figure 2. Screen used in "How the West was Won" game (from Burton & Brown, 1982).

parentheses or a short cut, for example, would improve the players position, and gives the student a chance to retake their turn. Thus the coach tries to expand the student's awareness of how they might play the game more successfully.

There are other examples of computer coaches that help students when they are carrying out tasks. For example, Anderson Boyle and Reiser (1985) have developed computer coaches for plane geometry and computer programming that pose problems to students and offer advice when the student takes a step in working the problem that reflects a common misunderstanding or error. Similarly, the PROUST system of Johnson and Soloway (1985) recognizes common errors students make in computer programming, and gives advice as to what the problem is. Computer coaches make it possible for students to spend their learning time actively carrying out tasks and projects, receiving personal help or guidance when they need it.

4 Reflection on Performance

Reflection refers to students looking back over what they did and analyzing their performance. There are four ways of permitting students to reflect on their performance (Collins & Brown, 1988): (1) imitation, (2) replay, (3) abstracted replay and (4) spatial reification. These can be illustrated in the context of tennis. Imitation occurs when a tennis coach shows you how you swing your racquet, perhaps contrasting it with the way you should swing it. Replay occurs when he videotapes your swing, which you can compare to videotapes of experts. An abstracted replay might be constructed by using reflective tape on critical points --the elbows, wrist, end of racquet, etc.-- so that the student can see how these points move with respect to each other in a videotaped replay. A spatial reification might be a plot of the trajectory of these points moving through space. In general students should be given multiple views, and be able to compare their performance to expert performance. Technology makes the last three forms of reflection possible, so this is a genuinely new teaching method emerging out of technology.

The benefits of reflection include:

a) What the student did becomes an object of study. The students begin to see their performance on tasks as data to be analyzed. They may never have taken what they did seriously before: reflection encourages them to think about their processes from the point of view of how they might be different, and what changes would lead to improved performance.

b) Students can compare their performance to others'. Reflection lets students see how different students and more expert performers carry out the same task. This encourages them to form hypotheses about what aspects of a process are critical to successful and unsuccessful performance.

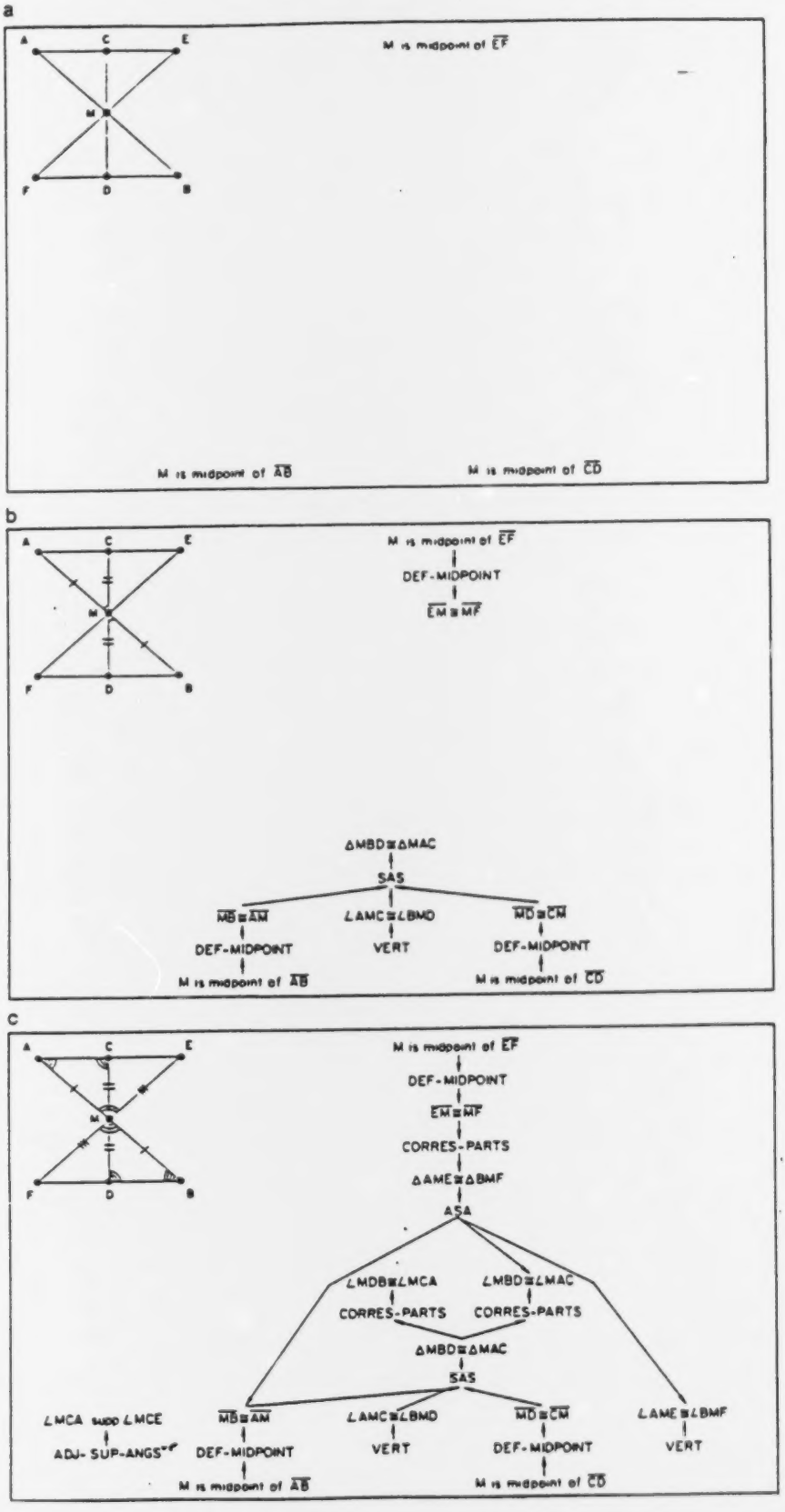
c) Abstractions about the process can be used for characterizing strategies. It is possible to describe various heuristics and metacognitive strategies (Schoenfeld 1985) in terms of the process the student is reflecting on. For example, in working geometry problems (see below), it is possible to characterize forward chaining as working from the givens and backward chaining as working from the statement to be proved. A good heuristic strategy is to start forward chaining from each of the givens to see what they imply, and then switch to backward chaining to work out the problem solution.

d) Spatial reification permits comparison of multiple performances to form abstractions. If students can see a process laid out in graphic form, then they can compare different people's approaches and try to characterize what aspects of the process are critical to expert vs. novice performance. The spatial representation permits them to see and even measure aspects of the process that are not apparent in a replay.

A good example of a spatial reification that permits reflection on a process is the problem solving trace that Anderson, Boyle and Reiser's (1985) Geometry Tutor constructs as the student works a problem. Figure 3 shows three views of the screen as a student works a problem. Initially in the top view the student sees a screen with the givens at the bottom and the statement to be proved at the top. In the middle screen the student has worked part way through the problem, working both forward from the givens and backwards from the statement to be proved. The third screen shows the final problem solution, along with various dead ends (e.g. <MDB=<MCA) that the student never used in the proof.

As Anderson, Boyle, Farrel, and Reiser (1984) point out, geometry proofs are usually presented in a fundamentally misleading way. Proofs on paper appear to be linear structures that start from a set of givens and proceed step by step (with a justification for each step) to the statement to be proved. But this is not how proofs are constructed by mathematicians or by anybody else. The process of constructing proofs involves an interplay between forward chaining from the givens and backward chaining from the goal statement. Yet, the use of paper and its properties encourage students to write proofs as if they were produced only by forward chaining--starting with the givens at the top of the page and working downward to the goal in a two column linear format (left column for the derived statements, right column for the logical justifications). If students infer that they should *construct* proofs this way, they will fail at any long proof. Properly designed computational learning environments can encourage students to proceed in both directions, moving forward, exploring the givens, and moving backwards, finding bridges to the goals.

The representation in Geometry Tutor is an abstraction of the problem solving process in terms of a "problem space". The system shows the states in the problem space that the student reached and the operators used to reach each of those states. Simply seeing the steps toward a



BEST COPY AVAILABLE

Fig. 3 (a) The Geometry Tutor's initial representation of the problem; (b) a representation in the middle of the problem; and (c) a representation at the solution of the problem. (from Anderson, Boyle, & Reiser, 1985)

solution reified in this way helps to create a problem space as a mental entity in its own right. This, in turn, makes it possible, for both teachers and students, to characterize problem-solving strategies in terms of abstractions that *refer* to the properties concretely manifested in the reified problem space. For example, in geometry it is a good strategy to forward chain at the beginning of a problem in order to understand the implications of the givens. Similarly, if you are stuck backward chaining, and do not see a way to connect your backward chain to any of the givens, then either go back to forward chaining or go back to the goal state again and try backward chaining along a different path.

Videotape first made systematic reflection on performance possible. The computer extends this to abstracted replays and reifications that highlight critical aspects of performance. In this way, students can analyze their performance from different perspectives and compare themselves to other students and experts.

5 Articulation

Articulation refers to methods for forcing students to explain and think about what they are doing i.e. making their tacit knowledge explicit. There are two ways that computers provide leverage for encouraging students to articulate their knowledge. First, computers make it possible for students to actually build their theories or ideas into artifacts that can be tested and revised. One challenge for their fellow students might be to show the limitations or failures of these theories. Second, computers can provide tools and settings where students try to articulate their ideas to other students, as in the Constitutional Convention convened on an electronic network in Boston (see Situated Learning).

The benefits of articulation include:

a) Making tacit knowledge explicit. When knowledge is tacit, it can only be used in contexts that elicit the knowledge automatically, i.e., that call up the knowledge because they are very similar to the conditions in which the knowledge was acquired. By forcing students to articulate their knowledge, it generalizes the knowledge from a particular context so that it can be used in other circumstances.

b) Making knowledge more available to be recruited in other tasks. Knowledge that is articulated as part of a set of interconnected ideas becomes more easily available. For example, if students acquire the idea of "thrashing" in problem solving (i.e. the concept of moving through a problem space without getting closer to the goal), then they can learn to recognize thrashing when they see it in different circumstances and develop strategies for dealing with it when it occurs.

c) Comparing strategies across contexts. When strategies are articulated, students can begin to see how the same strategies apply in different contexts. For example, the strategy of decomposing complex problems into simpler problems takes the form of writing subroutines in computer programming, and proving lemmas in geometry.

d) Articulation for other students promotes insight into alternative perspectives. If students try to explain an idea (or problem) to other students, then they begin to see the idea from the other students' perspectives. If they get responses from other students, they can see what difficulties other students have with the idea and how other people view the same issue (as will occur with the Constitutional Convention mentioned above).

We can illustrate articulation by a computer program called Robot Odyssey developed by the Learning Company. In it students first learn how to wire robots to behave in different ways when they sense different conditions. The robots have bumpers that detect when objects or walls are encountered on different sides, and sensors that detect the direction of particular objects in the vicinity of the robot. The robots have thrusters that can send them in different directions and grabbers to pick up objects. Students can wire these robots in very complicated patterns to move around, explore, and pick up or avoid different objects. Thus students can construct robots that behave in complicated ways, and turn them loose in the world to see what happens. The challenge of Robot Odyssey is to use the robots to navigate through a complicated maze, as in an Adventure game, where there are puzzles to solve and enemies to avoid in order to come through successfully. This then is the test of how well students have wired their robots, or more generally how deep their understanding is of how to build intelligent artifacts to deal with different situations.

Wiring robots is just one of many such enterprises. In languages like Logo (Goldenberg & Feurzeig, 1987) students can write programs that, for example, take any noun and generate a plural for it, or transform input sentences in active voice to passive voice. Thus students can articulate their grammatical theories in forms that can be tested out.

In an entirely different vein, Sharples (1980) has developed a Fantasy game kit that allows students to create computer Adventure games for other students to play (where they explore caves looking for treasure, avoiding monsters, and trying to find the way out). This forces students to articulate their images of different caverns in a series of caves, and pose problems for others that are solvable but challenging.

These examples illustrate some of the different ways computers can be used to foster articulation by students. While articulation is often encouraged by teachers in schools, computational articulation requires even more explicitness.

6 Exploration

Exploration involves pushing students to try out different hypotheses, methods and strategies to see their effects. This puts students in control of problem solving, but they need to learn how to explore productively. The computer provides powerful tools that allow students to explore hypotheses and solutions (i.e. problem spaces) faster, so they don't become frustrated.

The benefits of exploration include:

a) Learning how to set achievable goals. Many students set goals for themselves that are either too easily achievable or impossible to achieve (Atkinson, 1964). Studies of successful artists (Getzels & Csikszentmihalyi, 1976) have identified problem finding as the most critical skill for success. And yet very little effort in school goes into teaching students how to set reasonable goals and to revise their goals as they proceed deeper into a problem. Instead school emphasizes giving students well-defined tasks, unlike anything in the real world.

b) Learning how to form and test hypotheses. One of the major goals of inquiry teachers (Collins & Stevens, 1982, 1983) is to teach students how to formulate hypotheses, rules, or theories and then how to test out whether they are correct. Making sense out of the world around us requires these skills, and unless students practice forming and testing hypotheses in many different domains with the help of expert guidance, they will not learn to do so effectively.

c) Students will make discoveries on their own. When students are put into an environment where they are making and testing hypotheses, they get a sense of what it is like to be a scientist. They will feel the joy of generating their own ideas and seeing if they are correct. They may even discover genuinely novel ideas. But even when they come up with old ideas, they will at least sense where the ideas came from and why they are important.

We can illustrate the potential for exploration with the science laboratories developed by TERC (Mokros & Tinker, 1987). In one laboratory students can detect how far a moving object is from a sensor, and plot velocity, distance, and acceleration. This enables them to discover Galileo's laws of falling objects. Or they can conduct experiments with balls hitting other balls to see how changes in direction affect velocity. There are many different phenomena that students can investigate, and with the immediate graphic plotting they can test out many different ideas quickly.

Another example of an exploratory computer-based environment is the economics simulation called Smithtown developed by Shute and Bonar (1986). In Smithtown students can manipulate different variables, such as the price of coffee, to see how they affect other variables, such as the amount sold. They then are encouraged to try to figure out the various laws that

relate the different variables. Thus they are discovering basic economic relationships. Similarly students with a music or painting program can compose works quickly, and with much less effort. Hence they can explore many different techniques and see how effective they are. Computers provide powerful tools for exploration (Papert, 1980), and we are just beginning to investigate their potential.

7 Conclusion

As technology becomes cheaper it gives us the capability to realize resource intensive education once again. In particular, technology allows us to return to a kind of apprenticeship centered around modelling, coaching, and fading in situations that reflect the kinds of uses that the knowledge gained might be used. Moreover, technology enables us to create environments where new methods of learning - i.e., reflection, articulation, and exploration - are possible. This raises the question of what form education should take, given the capabilities and limitations of the technologies that are developing.

8 References

Anderson, J.R., Boyle, C.F., & Reiser, B.J. (1985). Intelligent tutoring systems. Science, 228, 456-468.

Anderson, J.R.; Boyle, C.F; Farrell, R.; & Reiser, B.J. (1984). Cognitive priciples in the design of computer tutors. In Proceedings of the Sixth Annual Conference of the Cognitive Science Society (pp. 2-9). Boulder, CO: University of Colorado.

Atkinson, J.W. (1964). An introduction to motivation. Princeton, NJ: Van Nostrand.

Brown, J.S., Burton, R.R., & deKleer, J. (1982). Padagogical natural language and knowledge engineering techniques in SOPHIE I, II, and III. In D. Sleeman and J.S. Brown (Eds.) Intelligent tutoring systems (pp. 227-282). New York, NY: Academic Press.

Burton, R.R. (1982). Dignosing bugs in a simple procedural skill. In D. Sleeman & J.S. Brown (Eds.). Intelligent Tutoring Systems (pp. 157-184). New York: Academic Press.

Burton, R.R. & Brown, J.S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman and J.S. Brown (Eds.) Intelligent tutoring systems (pp. 79-98). New York, NY: Academic Press.

Collins A. (1986). Teaching reading and writing with personal computers. In J. Orasanu (Ed.), A decade of reading research: Implications for practice (pp. 171-187). Hillsdale, NJ: Erlbaum.

Collins, A., & Brown, J.S. (in press). The computer as a tool for learning through reflection. In H. Mandl and Lesgold (Eds.), Learning issues for intelligent tutoring systems. New York: Springer.

Collins, A. & Smith, E. E. (1982) Teaching the process of reading comprehension. In D. K. Detterman & R. J. Sternberg (Eds.), How much and how can intelligence be increased? (pp. 173-185). Norwood, NJ: Ablex.

Collins, A., & Stevens, A. L. (1982). Goals and strategies of inquiry teachers. In R. Glaser (Ed.), Advances in instructional psychology (Vol. 2) (pp. 65-119). Hillsdale, NJ: Erlbaum.

Collins, A., & Stevens, A. L. (1983). A cognitive theory of interactive teaching. In C. M. Reigeluth (Ed.), Instructional design theories and models: An overview (pp. 247-278). Hillsdale, NJ: Erlbaum.

Collins, A., & Brown, J.S. (1988). The computer as a tool for learning through reflection. In H. Mandl and Lesgold (Eds.), Learning issues for intelligent tutoring systems. New York: Springer.

Collins, A., Brown, J.S., & Newman, S. (in press). Cognitive apprenticeship: Teaching the craft of reading, writing, and mathematics. In L.B. Resnick (Ed.), Cognition and Instruction: Issues and Agendas. Hillsdale, NJ: Erlbaum.

Dreyfus, H.L. & Dreyfus, S.E. (1986). Mind over machine: The power of human intuition and expertise in the era of the computer. New York: Fress Press.

Feurzeig, W. & White, B.Y. (1984). An articulate instrucitonal system for teaching arithmetic procedures. Cambridge, MA: Bolt Beranek and Newman, Inc.

Getzels, J., & Csikszentminhalyi, M. (1976). The creative vision: A longitudinal study of problem finding in art. New York: Wiley.

Goldenberg, E.P. & Feurzeig, W. (1987). Exploring language with Logo. Cambridge, MA: MIT Press.

Haertel, H. (1987). A qualitative approach to electricity. Palo Alto, CA: Xerox Corporation, Institute for Research on Learning.

Johnson, W.L. & Soloway, E. (1985). PROUST: An Automatic Debugger for Pascal Programs Byte, 10, 179-190.

Lampert, M. (1986). Knowing, doing, and teaching multiplication. Cognition and Instruction, 3, 305-342.

Mokros, J.R. & Tinker, R.F. (1987). The impact of microcomputer-based labs on children's ability to interpret graphs. Journal of Research in Science Teaching, 24, 369-383.

Palincsar, A.S., & Brown, A.L. (1984). Reciprocal teaching of comprehension-fostering and monitoring activities. Cognition and Instruction, 1, 117-175.

Papert, S. (1980), Mindstorms: Children, computers, and powerful ideas. New York: Basic Books.

Scardamalia, M., Bereiter, C., & Steinbach, R. (1984). Teachability of reflective processes in written composition. Cognitive Science, 8, 173-190.

Schoenfeld, A.H. (1983). Problem solving in the mathematics curriculum: A report,

recommendations and an annotated bibliography. The Mathematical Association of America, MAA Notes, No. 1.

Schoenfeld, A.H. (1985). Mathematical problem solving. New York: Academic Press.

Schoefeld, A.H. (in press). On mathematics as sense-making: An informal attack on the unfortunate divorce of formal and informal mathematics. In D.N. Perkins, J. Segal, & J. Voss Eds.) Informal reasoning and education. Hillsdale NJ: Erlbaum.

Sharples, M. (1980). A computer written language lab (DAI Working Paper No. 134). Edinburgh: University of Edinburgh, Scotland, Artificial Intelligence Department.

Shute, V. & Bonar, J. (1986). Intelligent tutoring systems for scientific inquiry skills. In Proceedings of the Eighth Annual Conference of the Cognitive Sciece Society (pp. 353-370). Hillsdale, NJ: Lawrence Erlbaum.

Snyder, T. & Palmer, J. (1986). In search of the most amazing thing: Children, education, and computers. Reading, MA: Addison-Wesley.

White, B.Y. and Frederiksen, J. (in press). Progressions of qualitative models as a foundation for intelligent learning environments. Artificial Intelligence.

ONR Distribution List -

Dr. Phillip L. Ackerman
University of Minnesota
Department of Psychology
75 East River Road
N218 Elliott Hall
Minneapolis, MN 55455

Dr. Beth Adelson
Department of Computer Science
Tufts University
Medford, MA 02155

AFOSR,
Life Sciences Directorate
Bolling Air Force Base
Washington, DC 20332

Dr. Robert Ahlers
Code N711
Human Factors Laboratory
Naval Training Systems Center
Orlando, FL 32813

Dr. John R. Anderson
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr. Stephen J. Andriole, Chairman
Department of Information Systems
and Systems Engineering
George Mason University
4400 University Drive
Fairfax, VA 22030

Technical Director, ARI
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Patricia Baggett
School of Education
610 E. University, Rm 1302D
University of Michigan
Ann Arbor, MI 48109-1259

Dr. Eva L. Baker
UCLA Center for the Study
of Evaluation
145 Moore Hall
University of California
Los Angeles, CA 90024

Dr. Gautam Biswas
Department of Computer Science
Box 1688, Station B
Vanderbilt University
Nashville, TN 37235

Dr. John Black
Teachers College, Box 8
Columbia University
525 West 120th Street
New York, NY 10027

Dr. Jeff Bonar
Learning R&D Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr. Robert Breaux
Code 7B
Naval Training Systems Center
Orlando, FL 32813-7100

Dr. John S. Brown
XEROX Palo Alto Research
Center
3333 Coyote Road
Palo Alto, CA 94304

Dr. John T. Bruer
James S. McDonnell Foundation
Suite 1610
1034 South Brentwood Blvd.
St. Louis, MO 63117

Dr. Bruce Buchanan
Computer Science Department
Stanford University
Stanford, CA 94305

LT COL Hugh Burns
AFHRL/IDI
Brooks AFB, TX 78235

Dr. Jaime G. Carbonell
Computer Science Department
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

BEST COPY A

ONR Distribution List -

Dr. Susan Carey
Department of Cognitive
and Neural Science
MIT
Cambridge, MA 02139

Dr. Pat Carpenter
Carnegie-Mellon University
Department of Psychology
Pittsburgh, PA 15213

Dr. John M. Carroll
IBM Watson Research Center
User Interface Institute
P.O. Box 704
Yorktown Heights, NY 10598

Dr. Fred Chang
Pacific Bell
2600 Camino Ramon
Room 3S-450
San Ramon, CA 94583

Dr. Davida Charney
English Department
Penn State University
University Park, PA 16802

Dr. Michelene Chi
Learning R & D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

Dr. Raymond E. Christal
UES LAMP Science Advisor
AFHRL/MOEL
Brooks AFB, TX 78235

Dr. William Clancey
Institute for Research
on Learning
3333 Coyote Hill Road
Palo Alto, CA 94304

Assistant Chief of Staff
for Research, Development,
Test, and Evaluation
Naval Education and
Training Command (N-5)
NAS Pensacola, FL 32508

Dr. Allan M. Collins
Bolt Beranek & Newman, Inc.
10 Moulton Street
Cambridge, MA 02238

Dr. Stanley Collyer
Office of Naval Technology
Code 222
800 N. Quincy Street
Arlington, VA 22217-5000

Dr. Albert T. Corbett
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Brian Dallman
Training Technology Branch
3400 TCHTW/TTGXC
Lowry AFB, CO 80230-5000

Dr. Denise DeLarosa
Psychology Department
Box 11A, Yale Station
Yale University
New Haven, CT 06520-7447

Dr. Andrea di Sessa
University of California
School of Education
Tolman Hall
Berkeley, CA 94720

Defense Technical
Information Center
Cameron Station, Bldg 5
Alexandria, VA 22314
Attn: TC
(12 Copies)

Dr. Richard Duran
Graduate School of Education
University of California
Santa Barbara, CA 93106

Dr. Susan Epstein
144 S. Mountain Avenue
Montclair, NJ 07042

ERIC Facility-Acquisitions
4350 East-West Hwy., Suite 1100
Bethesda, MD 20814-4475

AVAILABLE

ONR Distribution List

Dr. Marshall J. Farr, Consultant
Cognitive & Instructional Sciences
2520 North Vernon Street
Arlington, VA 22207

Dr. Paul Feltovich
Southern Illinois University
School of Medicine
Medical Education Department
P.O. Box 3926
Springfield, IL 62708

Mr. Wallace Feurzeig
Educational Technology
Bolt Beranek & Newman
10 Moulton St.
Cambridge, MA 02238

Dr. Craig I. Fields
ARPA
1400 Wilson Blvd.
Arlington, VA 22209

Dr. Gerhard Fischer
University of Colorado
Department of Computer Science
Boulder, CO 80309

Dr. Linda Flower
Carnegie-Mellon University
Department of English
Pittsburgh, PA 15213

Dr. Kenneth D. Forbus
University of Illinois
Department of Computer Science
1304 West Springfield Avenue
Urbana, IL 61801

Dr. Barbara A. Fox
University of Colorado
Department of Linguistics
Boulder, CO 80309

Dr. Carl H. Frederiksen
Dept. of Educational Psychology
McGill University
3700 McTavish Street
Montreal, Quebec
CANADA H3A 1Y2

Dr. John R. Frederiksen
BBN Laboratories
10 Moulton Street
Cambridge, MA 02238

Dr. Michael Friendly
Psychology Department
York University
Toronto ONT
CANADA M3J 1P3

Dr. Michael Genesereth
Stanford University
Computer Science Department
Stanford, CA 94305

Dr. Dedre Gentner
University of Illinois
Department of Psychology
603 E. Daniel St.
Champaign, IL 61820

Dr. Robert Glaser
Learning Research
& Development Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

Dr. Joseph Goguen
Computer Science Laboratory
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

Dr. Susan R. Goldman
Dept. of Education
University of California
Santa Barbara, CA 93106

Dr. Sherrie Gott
AFHRL/MOMJ
Brooks AFB, TX 78235-5601

Dr. T. Govindaraj
Georgia Institute of
Technology
School of Industrial
and Systems Engineering
Atlanta, GA 30332-0205

BEST COPY A

ONR Distribution List

Dr. James G. Greeno
School of Education
Stanford University
Room 311
Stanford, CA 94305

Dr. Dik Gregory
Admiralty Research
Establishment/AXB
Queens Road
Teddington
Middlesex, ENGLAND TW11OLN

Dr. Henry M. Halff
Halff Resources, Inc.
4918 33rd Road, North
Arlington, VA 22207

Dr. Bruce W. Hamill
Research Center
The Johns Hopkins University
Applied Physics Laboratory
Johns Hopkins Road
Laurel, MD 20707

Janice Hart
Office of the Chief
of Naval Operations
OP-111J2
Department of the Navy
Washington, D.C. 20350-2000

Dr. John R. Hayes
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh, PA 15213

Dr. Barbara Hayes-Roth
Knowledge Systems Laboratory
Stanford University
701 Welch Road
Palo Alto, CA 94304

Dr. Frederick Hayes-Roth
Teknowledge
P.O. Box 10119
1850 Embarcadero Rd.
Palo Alto, CA 94303

Dr. James Hollan
NPRDC, UCSD
Code 501
San Diego, CA 92152

Dr. Melissa Holland
Army Research Institute for the
Behavioral and Social Sciences
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Keith Holyoak
Department of Psychology
University of California
Los Angeles, CA 90024

Dr. Ed Hutchins
Intelligent Systems Group
Institute for
Cognitive Science (C-015)
UCSD
La Jolla, CA 92093

Dr. Alice M. Isen
Department of Psychology
University of Maryland
Catonsville, MD 21228

Dr. Janet Jackson
Rijksuniversiteit Groningen
Biologisch Centrum, Vleugel D
Kerklaan 30, 9751 NN Haren
The NETHERLANDS

Dr. Claude Janvier
Universite' du Quebec a Montreal
P.O. Box 8888, succ: A"
Montreal, Quebec H3C 3P8
CANADA

Dr. Robin Jeffries
Hewlett-Packard Laboratories, 3L
P.O. Box 10490
Palo Alto, CA 94303-0971

Dr. Marcel Just
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh, PA 15213

VAILABLE

ONR Distribution List -

Dr. Ruth Kanfer
University of Minnesota
Department of Psychology
Elliott Hall
75 E. River Road
Minneapolis, MN 55455

Dr. Wendy Kellogg
IBM T. J. Watson Research Ctr.
P.O. Box 704
Yorktown Heights, NY 10598

Dr. Dennis Kibler
University of California
Department of Information
and Computer Science
Irvine, CA 92717

Dr. David Kieras
Technical Communication Program
TIDAL Bldg., 2360 Bonisteel Blvd.
University of Michigan
Ann Arbor, MI 48109-2108

Dr. J. Peter Kincaid
Army Research Institute
Orlando Field Unit
c/o PM TRADE-E
Orlando, FL 32813

Dr. Janet L. Kolodner
Georgia Institute of Technology
School of Information
& Computer Science
Atlanta, GA 30332

Dr. Kenneth Kotovsky
Community College of
Allegheny County
808 Ridge Avenue
Pittsburgh, PA 15212

Dr. David H. Krantz
Department of Psychology
Columbia University
406 Schermerhorn Hall
New York, NY 10027

Dr. Pat Langley
University of California
Department of Information
and Computer Science
Irvine, CA 92717

Dr. Jill Larkin
Carnegie-Mellon University
Department of Psychology
Pittsburgh, PA 15213

Dr. Robert W. Lawler
Matthews 118
Purdue University
West Lafayette, IN 47907

Dr. Paul E. Lehner
Info. Sys. and Sys. Engr.
George Mason University
4400 University Drive
Fairfax, VA 22030

Dr. Alan M. Lesgold
Learning R&D Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr. Jim Levin
Department of
Educational Psychology
210 Education Building
1310 South Sixth Street
Champaign, IL 61820-6990

Dr. John Levine
Learning R&D Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr. Clayton Lewis
University of Colorado
Department of Computer Science
Campus Box 430
Boulder, CO 80309

Matt Lewis
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Library
Naval Training Systems Center
Orlando, FL 32813

Library
Naval War College
Newport, RI 02940

BEST COPY A

ONR Distribution List -

Science and Technology Division
Library of Congress
Washington, DC 20540

Dr. Marcia C. Linn
Graduate School
of Education, EMST
Tolman Hall
University of California
Berkeley, CA 94720

Dr. Don Lyon
P. O. Box 44
Higley, AZ 85236

Vern M. Malec
NPRDC, Code 52
San Diego, CA 92152-6800

Dr. Jane Malin
Mail Code EF5
NASA Johnson Space Center
Houston, TX 77058

Dr. Sandra P. Marshall
Dept. of Psychology
San Diego State University
San Diego, CA 92182

Dr. Manton M. Matthews
Department of Computer Science
University of South Carolina
Columbia, SC 29208

Dr. Richard E. Mayer
Department of Psychology
University of California
Santa Barbara, CA 93106

Dr. Joseph C. McLachlan
Code 52
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. James McMichael
Technical Director
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. D. Michie
The Turing Institute
George House
36 North Hanover Street
Glasgow G1 2AD
UNITED KINGDOM

Dr. George A. Miller
Dept. of Psychology
Green Hall
Princeton University
Princeton, NJ 08540

Dr. James R. Miller
MCC
3500 W. Balcones Center Dr.
Austin, TX 78759

Dr. Randy Mumaw
Training Research Division
HumRRO
1100 S. Washington
Alexandria, VA 22314

Dr. Allen Munro
Behavioral Technology
Laboratories - USC
1845 S. Elena Ave., 4th Floor
Redondo Beach, CA 90277

Dr. T. Niblett
The Turing Institute
George House
36 North Hanover Street
Glasgow G1 2AD
UNITED KINGDOM

Dr. Richard E. Nisbett
University of Michigan
Institute for Social Research
Room 5261
Ann Arbor, MI 48109

Dr. A. F. Norcio
Code 5530
Naval Research Laboratory
Washington, DC 20375-5000

Dr. Donald A. Norman
C-015
Institute for Cognitive Science
University of California
La Jolla, CA 92093

VAILABLE

ONR Distribution List

Deputy Technical Director
NPRDC Code 01A
San Diego, CA 92152-6800

Director, Training Laboratory,
NPRDC (Code 05)
San Diego, CA 92152-6800

Director, Manpower and Personnel
Laboratory,
NPRDC (Code 06)
San Diego, CA 92152-6800

Director, Human Factors
& Organizational Systems Lab,
NPRDC (Code 07)
San Diego, CA 92152-6800

Library, NPRDC
Code P201L
San Diego, CA 92152-6800

Technical Director
Navy Personnel R&D Center
San Diego, CA 92152-6800

Commanding Officer,
Naval Research Laboratory
Code 2627
Washington, DC 20390

Dr. Stellan Ohlsson
Learning R & D Center
University of Pittsburgh
Pittsburgh, PA 15260

Director of Research Programs
Office of Naval Research (Code 11)
800 North Quincy Street
Arlington, VA 22217-5000

Office of Naval Research
Code 1133
800 North Quincy Street
Arlington, VA 22217-5000

Office of Naval Research,
Code 1142
800 N. Quincy St.
Arlington, VA 22217-5000

Office of Naval Research,
Code 1142BI
800 N. Quincy Street
Arlington, VA 22217-5000

Office of Naval Research,
Code 1142CS
800 N. Quincy Street
Arlington, VA 22217-5000
(6 Copies)

Office of Naval Research,
Code 1142PS
800 N. Quincy Street
Arlington, VA 22217-5000

Office of Naval Research,
Code 11R
800 N. Quincy Street
Arlington, VA 22217-5000

Office of Naval Research,
Code 125
800 N. Quincy Street
Arlington, VA 22217-5000

Psychologist
Office of Naval Research
Branch Office, London
Box 39
FPO New York, NY 09510

Special Assistant for Marine
Corps Matters,
ONR Code 00MC
800 N. Quincy St.
Arlington, VA 22217-5000

Dr. Judith Orasanu
Basic Research Office
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Jesse Orlansky
Institute for Defense Analyses
1801 N. Beauregard St.
Alexandria, VA 22311

ONR Distribution List -

Dr. Douglas Pearce
1133 Sheppard W
Box 2000
Downsview, Ontario
CANADA M3H 3B9

Military Assistant for Training and
Personnel Technology,
OUSD (R & E)
Room 3D129, The Pentagon
Washington, DC 20301-3080

Dr. David N. Perkins
Project Zero
Harvard Graduate School
of Education
7 Appian Way
Cambridge, MA 02138

Dr. Steven Pinker
Department of Psychology
E10-018
MIT
Cambridge, MA 02139

Dr. Tjeerd Plomp
Twente University of Technology
Department of Education
P.O. Box 217
7500 AE ENSCHEDE
THE NETHERLANDS

Dr. Martha Polson
Department of Psychology
University of Colorado
Boulder, CO 80309-0345

Dr. Peter Polson
University of Colorado
Department of Psychology
Boulder, CO 80309-0345

Dr. Steven E. Poltrock
MCC
3500 West Balcones Center Dr.
Austin, TX 78759-6509

Dr. Harry E. Pople
University of Pittsburgh
Decision Systems Laboratory
1360 Scaife Hall
Pittsburgh, PA 15261

Dr. Joseph Psotka
ATTN: PERI-IC
Army Research Institute
5001 Eisenhower Ave.
Alexandria, VA 22333-5600

Dr. Lynne Reder
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr. Steve Reder
Northwest Regional
Educational Laboratory
400 Lindsay Bldg.
710 S.W. Second Ave.
Portland, OR 97204

Dr. James A. Reggia
University of Maryland
School of Medicine
Department of Neurology
22 South Greene Street
Baltimore, MD 21201

Dr. J. Wesley Regian
AFHRL/IDI
Brooks AFB, TX 78235

Dr. Fred Reif
Physics Department
University of California
Berkeley, CA 94720

Dr. Brian Reiser
Department of Psychology
Green Hall
Princeton University
Princeton, NJ 08540

Dr. Lauren Resnick
Learning R & D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15213

Dr. Gilbert Ricard
Mail Stop K02-14
Grumman Aircraft Systems
Bethpage, NY 11787

ONR Distribution List

Dr. J. Jeffrey Richardson
Center for Applied AI
College of Business
University of Colorado
Boulder, CO 80309-0419

Dr. Edwina L. Rissland
Dept. of Computer and
Information Science
University of Massachusetts
Amherst, MA 01003

Dr. Linda G. Roberts
Science, Education, and
Transportation Program
Office of Technology Assessment
Congress of the United States
Washington, DC 20510

Dr. William B. Rouse
Search Technology, Inc.
4725 Peachtree Corners Circle
Suite 200
Norcross, GA 30092

Dr. Roger Schank
Yale University
Computer Science Department
P.O. Box 2158
New Haven, CT 06520

Dr. Alan H. Schoenfeld
University of California
Department of Education
Berkeley, CA 94720

Dr. Janet W. Schofield
816 LRDC Building
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

Dr. Miriam Schustack
Code 52
Navy Personnel R & D Center
San Diego, CA 92152-6800

Dr. Colleen M. Seifert
Institute for Cognitive Science
Mail Code C-015
University of California, San Diego
La Jolla, CA 92093

Dr. Ben Shneiderman
Dept. of Computer Science
University of Maryland
College Park, MD 20742

Dr. Ted Shortliffe
Medical Computer Science Group
MSOB X-215
School of Medicine
Stanford University
Stanford, CA 94305-5479

Dr. Herbert A. Simon
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Robert L. Simpson, Jr.
DARPA/ISTO
1400 Wilson Blvd.
Arlington, VA 22209-2308

Dr. Derek Sleeman
Computing Science Department
King's College
Old Aberdeen AB9 2UB
Scotland
UNITED KINGDOM

Dr. Elliot Soloway
Yale University
Computer Science Department
P.O. Box 2158
New Haven, CT 06520

Dr. Richard C. Sorensen
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Kathryn T. Spoehr
Brown University
Department of Psychology
Providence, RI 02912

Dr. Robert J. Sternberg
Department of Psychology
Yale University
Box 11A, Yale Station
New Haven, CT 06520

BEST COPY A

ONR Distribution List

Dr. Kurt Steuck
AFHRL/MOA
Brooks AFB
San Antonio, TX 78235-5601

Dr. Albert Stevens
Bolt Beranek & Newman, Inc.
10 Moulton St.
Cambridge, MA 02238

Dr. Paul J. Sticha
Senior Staff Scientist
Performing Research Division
HumRRO
1100 S. Washington
Alexandria, VA 22314

Dr. John Tangney
AFOSR/NL, Bldg. 410
Bolling AFB, DC 20332-6448

Dr. Perry W. Thorndyke
FMC Corporation
Central Engineering Labs
1205 Coleman Avenue, Box 580
Santa Clara, CA 95052

Dr. Martin A. Tolcott
3001 Veazey Terr., N.W.
Apt. 1617
Washington, DC 20008

Dr. Douglas Towne
Behavioral Technology Labs
University of Southern California
1845 S. Elena Ave.
Redondo Beach, CA 90277

Headquarters, U. S. Marine Corps
Code MPI-20
Washington, DC 20380

Dr. Kurt Van Lehn
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr. Ralph Wachter
JHU-APL
Johns Hopkins Road
Laurel, MD 20707

Dr. Beth Warren
BBN Laboratories, Inc.
10 Moulton Street
Cambridge, MA 02238

Dr. Keith T. Wescourt
FMC Corporation
Central Engineering Labs
1205 Coleman Ave., Box 580
Santa Clara, CA 95052

Dr. Douglas Wetzel
Code 51
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Barbara White
BBN Laboratories
10 Moulton Street
Cambridge, MA 02238

Dr. Robert A. Wisher
U.S. Army Institute for the
Behavioral and Social Sciences
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Dr. Martin F. Wiskoff
Defense Manpower Data Center
550 Camino El Estero
Suite 200
Monterey, CA 93943-3231

Dr. Wallace Wulfeck, III
Navy Personnel R&D Center
Code 51
San Diego, CA 92152-6800

Dr. Masoud Yazdani
Dept. of Computer Science
University of Exeter
Prince of Wales Road
Exeter EX44PT
ENGLAND

Mr. Carl York
System Development Foundation
1 Maritime Plaza, #1770
San Francisco, CA 94111

AVAILABLE

Dr. Joseph L. Young
National Science Foundation
Room 320
1800 G Street, N.W.
Washington, DC 20550

Dr. Steven Zornetzer
Office of Naval Research
Code 114
800 N. Quincy St.
Arlington, VA 22217-5000

1988/05/01